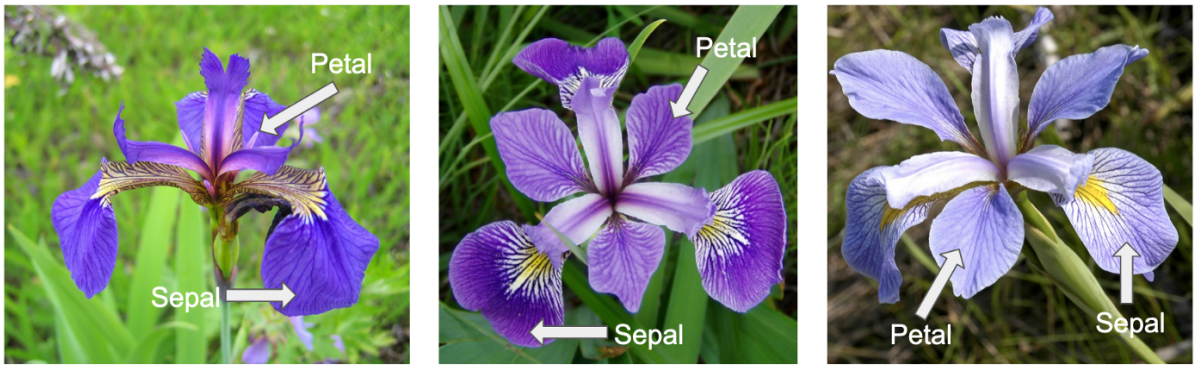
Objetivo.
Ejercitar el uso de árboles de decisión para problemas de clasificación.

Fuente de datos: Iris
Los datos consisten en 50 muestras de cada una de tres especies de flores Iris: *setosa, versicolor y virginica*. De cada flor se midieron 4 atributos: largo y ancho del sépalo y del pétalo.

Iris setosa | *Iris versicolor* | *Iris virginica*



En resumen, dada una instancia del dataset (es decir, los datos de una flor particular, una fila en el dataset sin la clase) queremos predecir a qué tipo de flor Iris pertenece.
Para cargar los datos pueden usar:

```
from sklearn import datasets
import pandas as pd

iris = datasets.load_iris()
iris_df = pd.DataFrame(iris['data'], columns=iris.feature_names)
```

Ejercicios

1. ¡Exploren los datos! Armen histogramas y otros gráficos que les permitan tener un conocimiento de los datos.
    a. ¿Pueden encontrar reglas de clasificación como las que usamos en la competencia con los datos del Titanic?
    b. ¿Cuántas instancias pudieron clasificar correctamente?
    c. ¿Con estos datos, es tan intuitivo encontrar reglas como en el ejemplo del Titanic?
    d. ¿Les parece que siempre podemos utilizar este método (armar reglas de manera manual) para predecir un valor de una fuente de datos?
2. Construyan un clasificador para predecir el tipo de flor utilizando el método de árboles de decisión de la biblioteca ***scikit-learn***.
3. Para el clasificador anterior, visualice el gráfico del árbol. ¿Cuál es el atributo utilizado como primer corte?

4. Realicen la comparación entre árboles que utilizan distintos criterios de corte (*entropy, gini gain, info gain*). ¿Cambia el atributo utilizado en el primer corte?
5. Armen árboles de decisión de distinta altura. Luego grafiquen y comparen cuantas instancias son clasificadas en cada una de sus hojas. ¿Qué árbol es más probable que sobreajuste? ¿Y cuál es más probable que subajuste?
   Importante: Recuerden separar previamente los datos en training/Testing. Pueden usar la siguiente función de Skit-Learn

```
from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test      =      train_test_split(iris_df,
iris.target, test_size = 0.1, random_state= 7)
```

6. En la diapositiva de la clase donde se presenta el algoritmo de Inducción Top-Down para árboles de decisión, se dejaron las siguiente preguntas (las volvemos a escribir a continuación). Responderlas.
   a. ¿Cómo hacemos el paso 2) para atributos continuos?
      ¿Cómo se modificaría el algoritmo?
   b. ¿Cómo se definen las regiones del paso 3) en ese caso?
7. Escriban un pseudocódigo del algoritmo que resuelva Inducción Top-Down para árboles de decisión.